UNITED STATES PATENT AND TRADEMARK OFFICE

H·A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/738,498	12/16/2003	Carlos V. Rozas	P17255	7850

25694 7590 11/01/2006

INTEL CORPORATION
P.O. BOX 5326
SANTA CLARA, CA 95056-5326

EXAMINER	
LE, CANH	

ART UNIT	PAPER NUMBER
2112	

DATE MAILED: 11/01/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/738,498	ROZAS, CARLOS V.	
	Examiner	Art Unit	
	Canh Le	2112	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *16 December 2003* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Drawings

The drawings are objected to because the response of module 230 of figure 2 should be labeled 220 to match its description in the specification at paragraph 0013, 0014, and 0016. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Specification

The disclosure is objected to because of the following informalities: Paragraph [0020] has a question about "Are we talking about OS kernel here?" for whom. Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-13 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claims are indefinite because it is unclear what is encompassed by the expression "baseline information" in claim 1. Therefore, the scope of the claims cannot be ascertained.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 26-37 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. A medium includes signal (see paragraph [0023]) and a signal is, per se, non-statutory.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –

Claims 20-25 are rejected under 35 U.S.C. 102(e) as being anticipated by Zimmer 4687 (20050114687 A1).

Claim 20

Zimmer 4687 discloses a system for monitoring software integrity, comprising:

a trusted computing device (paragraph [0028], line 4)

a protected partition machine running on the trusted computing device (figure 2, trusted partition 204 and protected memory 226);

a guest virtual machine running on the trusted computing device, the guest virtual machine including guest software (figure 2, Apps 216 and OS 216);

a secure memory area on the trusted computing device (figure 2, protected memory 226); and

an integrity monitor executing within the protected partition, the integrity monitor capable of generating a baseline hash value for the guest software initially, and a current hash value for the guest software during runtime. A secure virtual machine monitor (i.e., secure kernel) is equivalent to the integrity monitor. It runs at different operating modes or privilege levels of processors. It means to

execute in the protected partition (paragraph [0031], lines 18-19, paragraph [0032], lines 1-8; figure 2; paragraph [0040], lines 1-6).

Claim 21

Zimmer 4687 also discloses the system according to claim 20 wherein the secure memory area includes a trusted platform module ("TPM") (paragraph [0028], line 4).

Claim 22

Zimmer 4687 also disclose the system according to claim 20 wherein the trusted computing device may calculate a hash value for the integrity monitor and store the hash value for the integrity monitor in the secure memory area (paragraph [0049], lines 6-8).

Claim 23

Zimmer 4687 also disclose The system according to claim 22 wherein the hash value for the integrity monitor may be used to verify the integrity monitor prior to enabling the integrity monitor to access the baseline hash value stored in the secure memory area. (figure 2; paragraph [0032], lines 10-13).

Claim 24

Zimmer 4687 also disclose the system according to claim 21 wherein the trusted computing device executes in Secure Execution Machine ("SMX") mode and the secure memory area includes one of the TPM and a designated non-writable

memory area. A resource protected list can be stored in a TPM, PCR (paragraph [0049], lines 4-9). A protected firmware resource 106 is stored in the protected memory 226 (i.e., a non-writable memory) (figure 1,figure 2; paragraph [0041], lines 1-12).

Claim 25

Zimmer 4687 also disclose the system according to claim 24 wherein a secure launch module may calculate a hash value for the integrity monitor (paragraph [0029], lines 7-9) and store the hash value for the integrity monitor in the secure memory area (paragraph [0029]).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.

4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zimmer 1968 (US 20050021968 A1) in view of Zimmer 4687 (20050114687 A1).

Claim 1

Zimmer 1968 discloses a method of monitoring software executing on a trusted computing device comprising:

storing the baseline information in a secure memory area. A baseline level of security is equivalent to a baseline information (paragraph [0019], lines 1-6).

processing the guest software during runtime according to a predefined methodology to determine current runtime information. A firmware[1] is equivalent to software in a context of the prior. The firmware update drive issues a SENDER command upon execution (i.e. runtime). A processor hash-extends a binary image of update driver using hash algorithm SHA-1 (i.e. predefined methodology) (paragraph [0050])

comparing the current runtime information to the baseline information stored in the secure memory area to determine whether the guest software has been compromised (paragraph [0035], lines 7-9).

Zimmer 1968 does not disclose generating in a protected partition on the trusted computing device baseline information.

Zimmer 4687 discloses generating in a protected partition on the trusted computing device baseline information pertaining to guest software in a guest

virtual machine. A protected operating partition is equivalent to a trusted partition (paragraph [0031], lines 11-19). A code in trusted partition can be setup to run at different operating modes or privilege levels of processor (paragraph [0032], lines 1-3). As a result, some metadata information are generated from this protected partition relate to guest software. Virtual machine monitor (VMM) presents to other software (i.e. "guest" software) the abstraction of one or more virtual machines (VMs). In figure 2, Secure virtual machine monitor (SVMM) 110 presents other software (e.g., OS 216 or 212 APPS) as guest software.

Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the method of Zimmer 1968 by including the step of generating in a protected partition on the trusted computing device baseline information because it would perform system resource verification tests to ensure proper resource configuration/operation (paragraph [0005], lines 13-14, Zimmer 4687).

[1] Definition of firmware on page 438 of IEEE 100, THE AUTHORITATIVE DICTIONARY OF IEEE STANDARDS TERMS, SEVEN EDITION, IEEE Published by Standards Information Network IEEE Press, ISBN 0-7381-2601-2

Claim 2

Zimmer 1968 also does not disclose performing a hash function on the guest software to obtain a hash value.

Zimmer 4687 discloses performing a hash function on the guest software to obtain a hash value (paragraph [0029], lines 1-4). A firmware is a software. The

pre-boot firmware 102 in the pre-boot environment 100 based on the content descriptors may include any firmware resource. (paragraph 0022], lines 4-6). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the method of Zimmer 1968 by including the step of performing a hash function on the guest software to obtain a hash value because it would provide a protection for software.

Claim 3

Zimmer 1968 also does not disclose performing a hash function on one of each component of the guest software and a collection of components of the guest software

Zimmer 4687 discloses performing the hash function on the guest software includes performing a hash function on one of each component of the guest software and a collection of components of the guest software. (paragraph [0029], lines 1-3, paragraph [0049], lines 19-20). Each protection descriptor is equivalent to each component of the guest software and descriptors stored in the resource protection list (RPL) 108 are equivalent to a collection of component of the guest software. Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the method of Zimmer 1968 by including the step of performing the hash function on the guest software includes performing a hash function on one of each component of the guest software and a collection of components of the guest software because it would provide a protection for software.

Claim 4

Zimmer 1968 also discloses the method according to claim 2 wherein performing the hash function on the guest software to obtain the hash value further comprises at least one of performing the hash function on the guest software prior to execution to obtain an initial static baseline value and performing the hash function on the guest software immediately upon execution to obtain an initial runtime baseline value. A cryptographic hash is equivalent to hash function prior to execution (paragraph [0032], lines 1-3). Performing the hash function on the guest software immediately upon execution to obtain an initial runtime baseline is equivalent to performing a hash operation on the firmware update driver (paragraph [0025], lines 5-6).

Claim 5

Zimmer 1968 also discloses the method according to claim 4 wherein processing the guest software during runtime according to a predefined methodology further comprises performing the hash function periodically on the guest software during runtime to obtain a current hash value. Performing the hash function periodically on the guest software is equivalent to performing a hash operation on the firmware update driver (paragraph [0025], lines 5-6).

Claim 6

Zimmer 1968 also discloses the method according to claim 5 wherein comparing the current runtime information to the baseline information further comprises

comparing the current hash value to the baseline hash value. The current hash value to the baseline hash value is equivalent to current platform environment with the given platform environment (paragraph [0035], lines 7-9).

Claim 7

Zimmer 1968 also discloses the method according to claim 1 wherein generating the baseline information comprises retrieving the baseline information from a storage location on the trusted computing device. Retrieving the baseline information is equivalent to retrieve key and certificate data (paragraph [0015], lines 5-9). A trusted computing device is equivalent to a TPM 126 (paragraph [0018]).

Claim 8

Zimmer 1968 also discloses the method according to claim 1 wherein storing the baseline information in the secure memory area further comprises storing the hash value in a trusted platform module ("TPM") (paragraph [0019], lines 1-6).

Claim 9

Zimmer 1968 also does not disclose performing a secure launch of the trusted computing platform prior to generating the baseline information.
Zimmer 4687 discloses performing a secure launch of the trusted computing platform prior to generating the baseline information (paragraph [0050], lines 13-15; paragraph [0052], lines 6-8). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the

method of Zimmer 1968 by including the step of performing a secure launch of the trusted computing platform prior to generating the baseline information because it would provide for future authentication or validation of trustworthiness.

Claim 10

Zimmer 1968 also does not disclose storing the hash value in one of a TPM and a designated non-writable memory area.

Zimmer 4687 discloses storing the baseline information in the secure memory area further comprises storing the hash value in one of a TPM and a designated non-writable memory area. Storing the hash code in a secure register such as a TPM PCR. (paragraph [0049], lines 7-8). A protected memory is equivalent to non-writable memory (paragraph [0041], lines 10-17). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the method of Zimmer 1968 by including the step of storing the hash value in one of a TPM and a designated non-writable memory area because it would be protected from malignant modifications.

Claim 11

Zimmer 1968 also does not disclose executing at least a portion of the guest software in a designated non-writable memory area.

Zimmer 4687 discloses executing at least a portion of the guest software in a designated non-writable memory area. A protected memory is equivalent to non-writable memory (paragraph [0041], lines 10-17). Same motivation as claim 10.

Claim 12

Zimmer 1968 also discloses the method according to claim 1 wherein the predefined methodology includes at least one of a checksum, MD5 and SHA1. (paragraph [0050], lines 5-7).

Claim 13

Zimmer 1968 also does not disclose the protected partition which includes a root virtual machine.

Zimmer 4687 discloses the protected partition which includes a root virtual machine. A trusted partition (i.e., a protected operating partition) is equivalent to the protected partition. The trusted partition includes a secure OS and a secure virtual machine monitor (SVMM) (figure2; paragraph [0026], lines 5-7; paragraph [0031], lines 18-19). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the method of Zimmer 1968 by including the step of the protected partition which includes a root virtual machine because a trusted protected (i.e., a protected operating partition) includes a root virtual machine.

Claims 14-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zimmer 1968 (US 20050021968 A1) in view of Zimmer 4687 (20050114687 A1).

Claim 14

Zimmer 1968 discloses a method of monitoring the integrity of a trusted computing device, comprising:

storing the baseline value in a secure memory area; paragraph [0019], lines 1-6);

the integrity monitor periodically processing the guest software while executing to generate a current hash value. (paragraph [0025], lines 5-6).) and the integrity monitor comparing the baseline hash value in the secure memory area to the current hash value to determine whether the guest software has been compromised (paragraph [0035], lines 7-9).

Zimmer 1968 does not disclose launching a protected partition and a guest virtual machine on the trusted computing device, executing an integrity monitor in the protected partition and guest software in the guest virtual machine, and processing the guest software in the guest virtual machine.

Zimmer 4687 discloses launching a protected partition and a guest virtual machine on the trusted computing device. A protected partition is equivalent to a trusted partition (i.e. a protected operating partition). The trusted partition may be configured (i.e., set up) to run at different operating modes or privilege levels of processors. It means for the protected partition and guest virtual machine (paragraph [0031], lines 18-19; paragraph [0032], lines 1-8; figure 2);

executing an integrity monitor in the protected partition and guest software in the guest virtual machine. A secure virtual machine monitor (i.e., secure kernel) is equivalent to the integrity monitor. It runs at different operating modes or privilege levels of processors. It means to execute in the protected partition and guest software in the guest virtual machine (paragraph [0031], lines 18-19, paragraph [0032], lines 1-8; figure 2; paragraph [0040], lines 1-6).

the integrity monitor processing the guest software in the guest virtual machine to generate a baseline hash value. A trusted platform module (TPM) includes a secure encryption function in environment 101(paragraph [0029], lines 1-3 and lines 9-12). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the method of Zimmer 1968 by including the step of launching a protected partition and a guest virtual machine, executing an integrity monitor in the protected partition and guest software, and processing the guest software because it would perform system resource verification tests to ensure proper resource configuration/operation (paragraph [0005], lines 13-14, Zimmer 4687).

Claim 15

Zimmer 1968 does not disclose storing the baseline value in a secure memory area including storing the baseline value in at least one of a trusted platform module ("TPM") and a designated non-writable memory area.

Zimmer 4687 discloses the method according to claim 14 wherein storing the baseline value in a secure memory area includes storing the baseline value in at least one of a trusted platform module ("TPM") and a designated non-writable memory area. A resource protected list can be stored in a TPM, PCR (paragraph [0049], lines 4-9). A protected firmware resource 106 is stored in the protected memory 226 (i.e., a non-writable memory) (figure 1, figure 2; paragraph [0041], lines 1-12). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the method of Zimmer 1968

by including the step of storing the baseline value in a secure memory area includes storing the baseline value in at least one of a trusted platform module ("TPM") and a designated non-writable memory area because it would because it would be protected from malignant modifications.

Claim 16

Zimmer 1968 also discloses processing and storing a value corresponding to the integrity monitor (paragraph [0035], lines 5-6).

Claim 17

Zimmer 1968 also discloses the method according to claim 16 further comprising verifying the integrity monitor prior to comparing the baseline hash value to the current hash value. An integrity metric is generated by performing a hash operation on the firmware (paragraph [0025], lines 3-6). The current hash value to the baseline hash value is equivalent to current platform environment with the given platform environment (paragraph [0035], lines 7-9).

Claim 18

Zimmer 1968 does not disclose processing the guest software in the guest virtual machine to generate the baseline hash value includes retrieving the baseline hash value from a storage location.

Zimmer 4687 also discloses processing the guest software in the guest virtual machine to generate the baseline hash value includes retrieving the baseline hash value from a storage location (paragraph [0029], lines 11-12).

Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the method of Zimmer 1968 by including the step of generating in a protected partition on the trusted computing device baseline information because it would perform system resource verification tests to ensure proper resource configuration/operation (paragraph [0005], lines 13-14, Zimmer 4687).

Claim 19

Zimmer 1968 does not disclose launching a protected partition includes launching a root virtual machine.

Zimmer 4687 discloses launching a protected partition includes launching a root virtual machine. A trusted partition (i.e., a protected operating partition) is equivalent to the protected partition. The trusted partition includes a secure OS and a secure virtual machine monitor (SVMM) (figure 2; paragraph [0026], lines 5-7; paragraph [0031], lines 18-19). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the method of Zimmer 1968 by including the step of launching a protected partition includes launching a root virtual machine because a trusted protected (i.e., a protected operating partition) includes a root virtual machine.

Claims 26-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zimmer 1968 (US 20050021968 A1) in view of Zimmer 4687 (20050114687 A1).

Claim 26

Zimmer 1968 discloses an article comprising a medium accessible by a trusted computing device, the medium having stored thereon instructions that, when executed by the trusted computing device, cause the trusted computing device to (paragraph [0058]):

store the baseline information in a secure memory area. A baseline level of security is equivalent to a baseline information (paragraph [0019], lines 1-6).

process the guest software during runtime according to a predefined methodology to determine current runtime information. A firmware[1] is equivalent to software in a context of the prior. The firmware update drive issues a SENDER command upon execution (i.e. runtime). A processor hash-extends a binary image of update driver using hash algorithm SHA-1 (i.e. predefined methodology) (paragraph [0050])

compare the current runtime information to the baseline information stored in the secure memory area to determine whether the guest software has been compromised (paragraph [0035], lines 7-9).

Zimmer 1968 does not disclose generating in a protected partition on the trusted computing device baseline information.

Zimmer 4687 discloses generating in a protected partition on the trusted computing device baseline information pertaining to guest software in a guest virtual machine. A protected operating partition is equivalent to a trusted partition (paragraph [0031], lines 11-19). A code in trusted partition can be setup to run at different operating modes or privilege levels of processor (paragraph [0032], lines

1-3). As a result, some metadata information are generated from this protected partition relate to guest software. Virtual machine monitor (VMM) presents to other software (i.e. "guest" software) the abstraction of one or more virtual machines (VMs). In figure 2, secure virtual machine monitor (SVMM) 110 presents other software (e.g., OS 216 or 212 APPS) as guest software.

Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the article of Zimmer 1968 by including a machine-readable medium because it would store or transmit information in a form of readable by a machine.

[1] Definition of firmware on page 438 of IEEE 100, THE AUTHORITATIVE DICTIONARY OF IEEE STANDARDS TERMS, SEVEN EDITION, IEEE Published by Standards Information Network IEEE Press, ISBN 0-7381-2601-2

Claim 27

Zimmer 1968 also does not disclose performing a hash function on the guest software to obtain a hash value.

Zimmer 4687 discloses performing a hash function on the guest software to obtain a hash value (paragraph [0029], lines 1-4). A firmware is a software. The pre-boot firmware 102 in the pre-boot environment 100 based on the content descriptors may include any firmware resource. (paragraph 0022], lines 4-6). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the article of Zimmer 1968 by including

the step of performing a hash function on the guest software to obtain a hash value because it would provide a protection for software.

Claim 28

Zimmer 1968 also does not disclose performing a hash function on one of each component of the guest software and a collection of components of the guest software

Zimmer 4687 discloses performing the hash function on the guest software includes performing a hash function on one of each component of the guest software and a collection of components of the guest software. (paragraph [0029], lines 1-3, paragraph [0049], lines 19-20). Each protection descriptor is equivalent to each component of the guest software and descriptors stored in the resource protection list (RPL) 108 are equivalent to a collection of component of the guest software. Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the article of Zimmer 1968 by including the step of performing the hash function on the guest software includes performing a hash function on one of each component of the guest software and a collection of components of the guest software because it would provide a protection for software.

Claim 29

Zimmer 1968 also discloses the method according to claim 2 wherein performing the hash function on the guest software to obtain the hash value further comprises at least one of performing the hash function on the guest software

prior to execution to obtain an initial static baseline value and performing the hash function on the guest_software immediately upon execution to obtain an initial runtime baseline value. A cryptographic hash is equivalent to hash function prior to execution (paragraph [0032], lines 1-3). Performing the hash function on the guest software immediately upon execution to obtain an initial runtime baseline is equivalent to performing a hash operation on the firmware update driver (paragraph [0025], lines 5-6).

Claim 30

Zimmer 1968 also discloses the method according to claim 4 wherein processing the guest software during runtime according to a predefined methodology further comprises performing the hash function periodically on the guest software during runtime to obtain a current hash value. Performing the hash function periodically on the guest software is equivalent to performing a hash operation on the firmware update driver (paragraph [0025], lines 5-6).

Claim 31

Zimmer 1968 also discloses the method according to claim 5 wherein comparing the current runtime information to the baseline information further comprises comparing the current hash value to the baseline hash value. The current hash value to the baseline hash value is equivalent to current platform environment with the given platform environment (paragraph [0035], lines 7-9).

Claim 32

Zimmer 1968 also discloses the method according to claim 1 wherein generating the baseline information comprises retrieving the baseline information from a storage location on the trusted computing device. Retrieving the baseline information is equivalent to retrieve key and certificate data (paragraph [0015], lines 5-9). A trusted computing device is equivalent to a TPM 126 (paragraph [0018]).

Claim 33

Zimmer 1968 also discloses the method according to claim 1 wherein storing the baseline information in the secure memory area further comprises storing the hash value in a trusted platform module ("TPM") (paragraph [0019], lines 1-6).

Claim 34

Zimmer 1968 also does not disclose performing a secure launch of the trusted computing platform prior to generating the baseline information.
Zimmer 4687 discloses performing a secure launch of the trusted computing platform prior to generating the baseline information (paragraph [0050], lines 13-15; paragraph [0052], lines 6-8). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the article of Zimmer 1968 by including the step of performing a secure launch of the trusted computing platform prior to generating the baseline information because it would provide for future authentication or validation of trustworthiness.

Claim 35

Zimmer 1968 also does not disclose storing the hash value in one of a TPM and a designated non-writable memory area.

Zimmer 4687 discloses storing the baseline information in the secure memory area further comprises storing the hash value in one of a TPM and a designated non-writable memory area. Storing the hash code in a secure register such as a TPM PCR. (paragraph [0049], lines 7-8). A protected memory is equivalent to non-writable memory (paragraph [0041], lines 10-17). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the article of Zimmer 1968 by including the step of storing the hash value in one of a TPM and a designated non-writable memory area because it would be protected from malignant modifications.

Claim 36

Zimmer 1968 also does not disclose executing at least a portion of the guest software in a designated non-writable memory area.

Zimmer 4687 discloses executing at least a portion of the guest software in a designated non-writable memory area. A protected memory is equivalent to non-writable memory (paragraph [0041], lines 10-17). Same motivation as claim 10.

Claim 37

Zimmer 1968 also does not disclose the protected partition which includes a root virtual machine.

Zimmer 4687 discloses the protected partition which includes a root virtual machine. A trusted partition (i.e., a protected operating partition) is equivalent to the protected partition. The trusted partition includes a secure OS and a secure virtual machine monitor (SVMM) (figure2; paragraph [0026], lines 5-7; paragraph [0031], lines 18-19). Thus, it would have been obvious to the person of ordinary skill in the art at the time of the invention was made to modify the article of Zimmer 1968 by including the step of the protected partition which includes a root virtual machine because a trusted protected (i.e., a protected operating partition) includes a root virtual machine.

Conclusion

The prior arts made of record and not relied upon are considered pertinent to applicant's disclosure. The prior arts disclose method for performing a trusted firmware/BIOS update, protected partition, TPM, virtual machine operation, virtual translation, virtual machine monitor, tamper detection, Multiple trusted computing environment, system and method for resetting a platform configuration register, and Lagrande Technology Architecture Overview.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Canh Le whose telephone number is 571-270-1380. The examiner can normally be reached on Monday to Friday 7:30AM to 5:00PM other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Walter Griffin can be reached on 571-272-1447. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Canh Le
October 23, 2006

WALTER D. GRIFFIN
SUPERVISORY PATENT EXAMINER